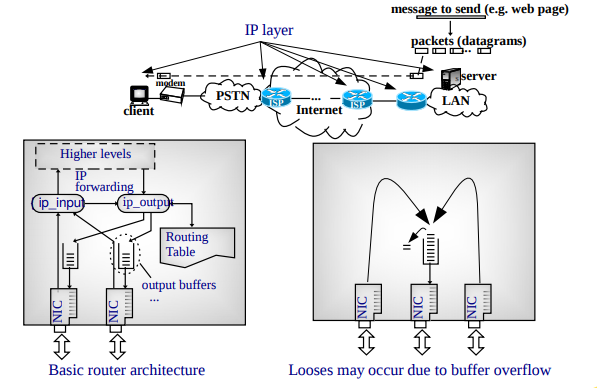
**Unit 2: IP Networks**

**IP layer service**

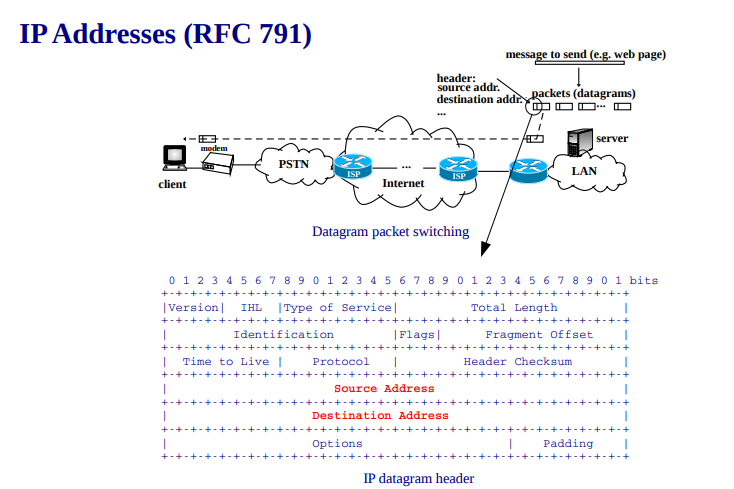
Internet protocol (IP) goal is routing datagrams. IP main design goal was interconnecting hosts attached to LANs/WANs networks of different technologies.

IP characteristics:

* Connectionless.
* Stateless.
* Best effort.

****

**IP addresses**

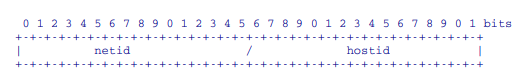
****

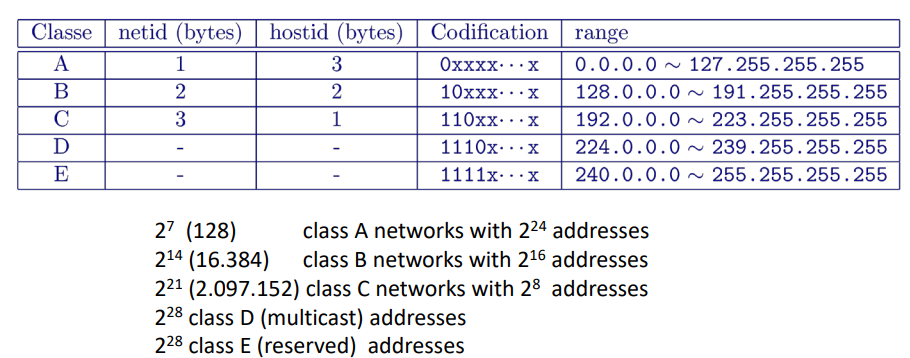
Characteristics:

* 32 bits (4 bytes).
* Dotted point notation: four bytes in decimal, e.g., 147.83.24.28.
* Netid identifies the network.
* Hostid identifies the host within the network.
* An IP address identifies an interface: an attachment point to the network.
* All IP addresses in Internet must be different. To achieve this goal, IANA assign address blocs to Regional Internet Registries, RIR. RIR assign addresses to ISPs, and ISPs to their customers.

Classes:

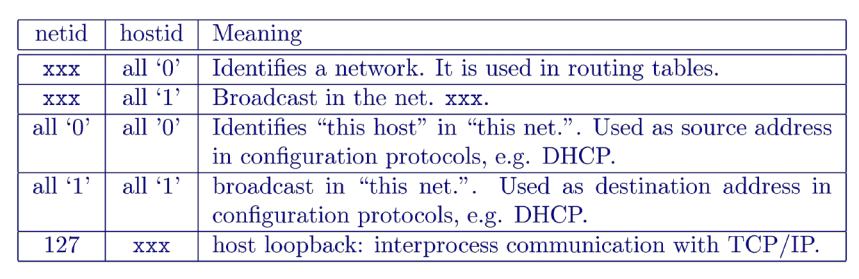
* The highest bits identify the class.
* The number of IP bits of netid/hostid varies in classes A/B/C.
* D class is for multicast addresses (e.g., 224.0.0.2: “all routers”).
* E class are reserved addresses.

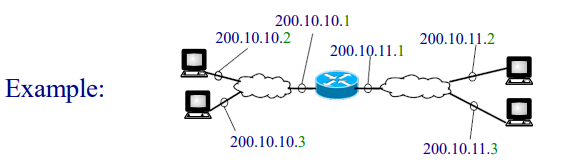


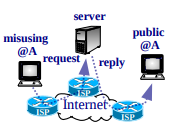


Special addresses:

* Special addresses cannot be used for a physical interface.
* Each network has two special addresses: network and broadcast addresses.



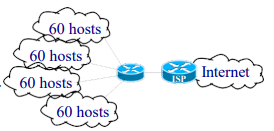


Private addresses (RFC 1918 Y1996):

* Commercial Oss include the TCP/IP stack.
* TCP/IP is used to network many kinds of electronic devices (such as PC, printer, laptop, PDA, IP camera, DVD player, GPS, balance…).
* Addresses assigned to RIRs by IANA are called public, global or registered.
* What if we arbitrarily assign a registered address to a host?
  + It may be filtered by our ISP or cause trouble to the right host using that address.
* Private addresses have been reserved for devices not using public addresses. These addresses are not assigned to any RIR (are not unique). There are addresses in each class:
  + 1 class A network: 10.0.0.0.
  + 16 class B networks: 172.16.0.0 ~ 172.31.0.0.
  + 256 class C networks: 192.168.0.0 ~ 192.168.255.0.

**Subnetting (RFC 950 Y1985)**

Initially the netid was given by the address class: A with 224 addresses, B with 216 addresses and C with 28 addresses. What if we want to divide the network?



Subnetting allows adding bits from the hostid to the netid (called subnetid bits).

Example: For the ISP the network prefix is 24 bits. For the internal router the network prefix is 26 bits. The 2 extra bits allows 4 “subnetworks”.

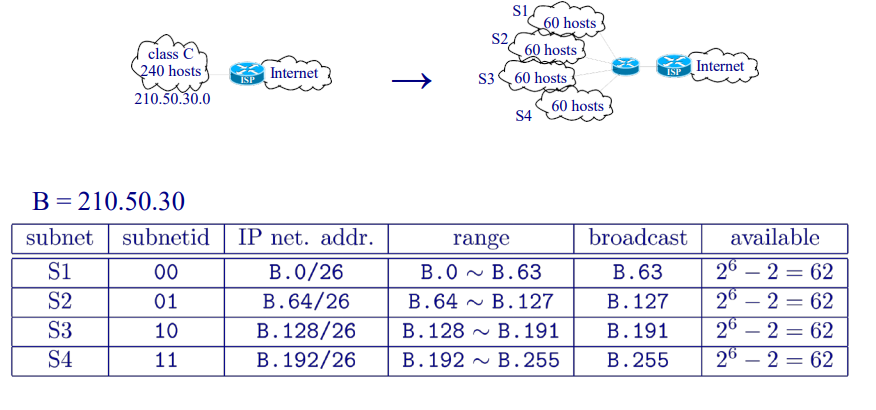
A mask is used to identify the size of the netid+subnetid prefix.

Mask notations:

* Dotted, as 255.255.255.192.
* Giving the mask length (number of bits) as 210.50.30.0/26 (26 bits of ones consecutives and 6 bits of 0, that is equal to 255.255.255.192).

Example:

We want to subnet the address 210.50.30.0/24 in 4 subnets.

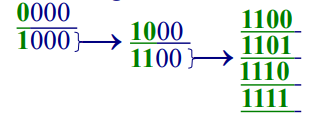


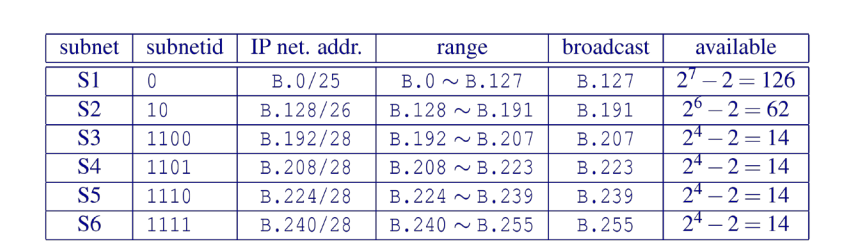
Variable Length Subnet Mask (VLSM):

Subnetworks of different sizes.

Example, subnetting a class C address:

* We have 1 byte for subnetid + hostid.
* Subnetid is green, chosen subnets addresses are underlined.



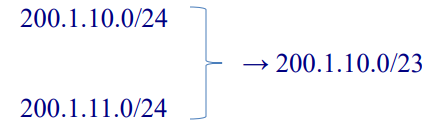


We lost 3 addresses in each subnet (the broadcast, the network and the router).

Classless Inter-Domain Routing, CIDR (RFC 1519 Y1993):

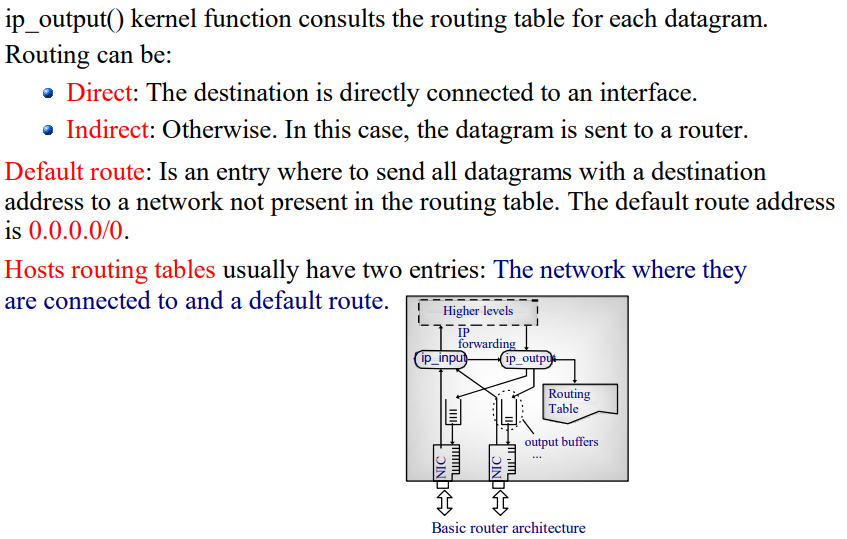
Initially, Internet backbone routing tables did no use masks: netid was derived from the IP address class (default /8, /16, /24). When the number of networks in Internet started growing exponentially, routing tables size started exploding. In order to reduce routing tables size, CIDR proposed a “rational” geographical-based distribution of IP addresses to be able to “aggregate routes”, and use masks instead of classes.

Aggregation example:

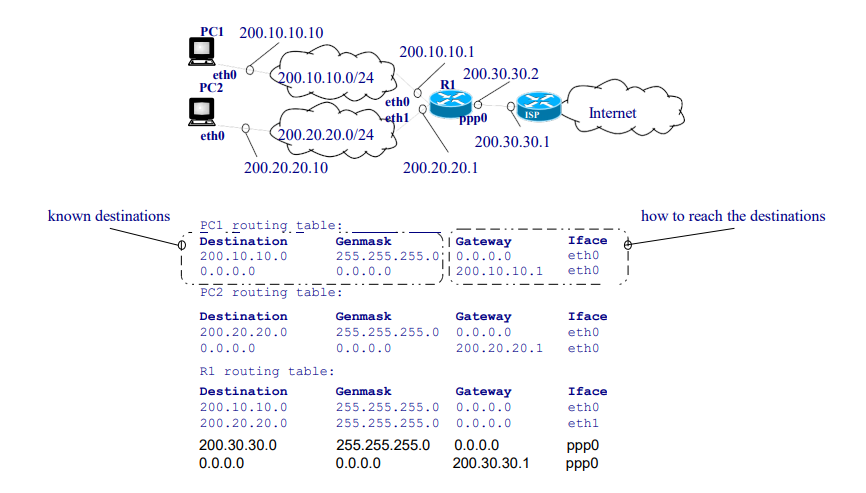


The term summarization is normally used when aggregation is done at a class boundary (e.g., a group of subnets is summarized with their classful base address).

**Routing tables**

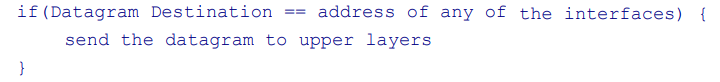
****

Unix Example:

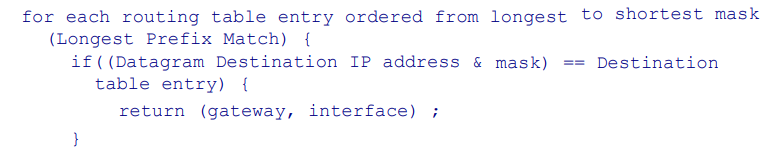


Datagram delivery Algorithm:

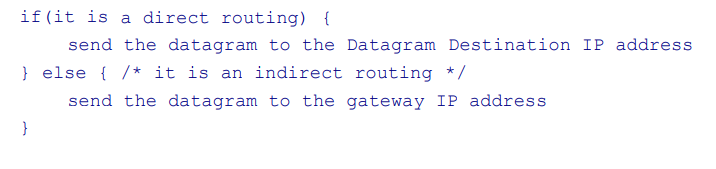
1. Check if the device itself is the destination:



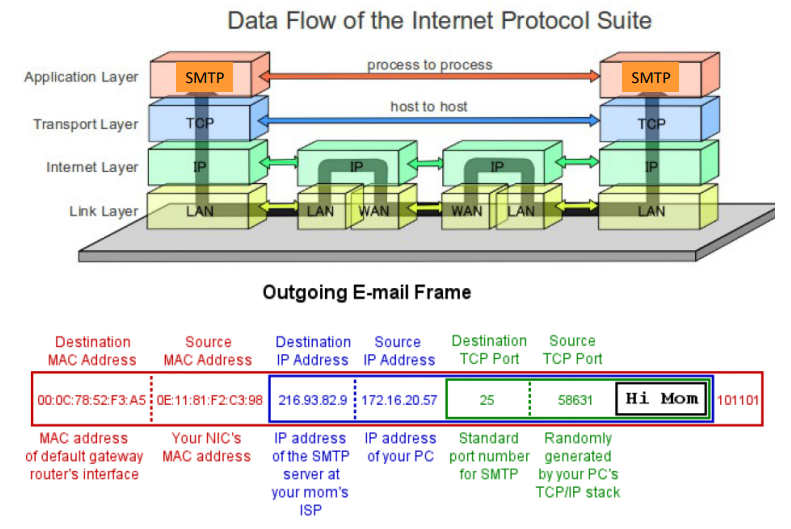
1. Consult the routing table:



1. Forward the datagram:



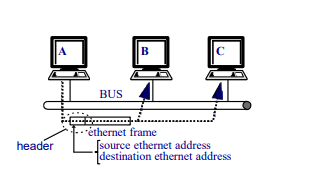
**ARP protocol**

****

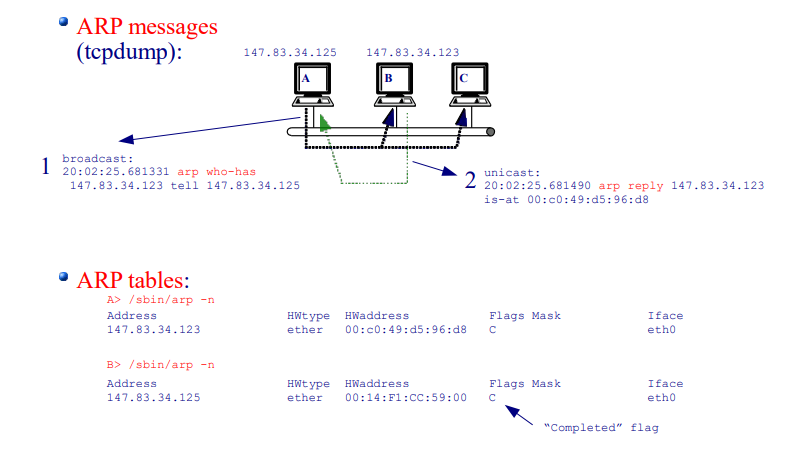
Address Resolution Protocol, ARP (RFC 826 Y1982):

To send the datagram, IP layer may have to pass a “physical address” to the NIC driver. Physical addresses are also called MAC or hardware addresses. ARP translate IP addresses to “physical addresses” (used by the physical network). If needed, IP calls ARP module to obtain the “physical addresses” before the NIC driver call.

Ethernet example:



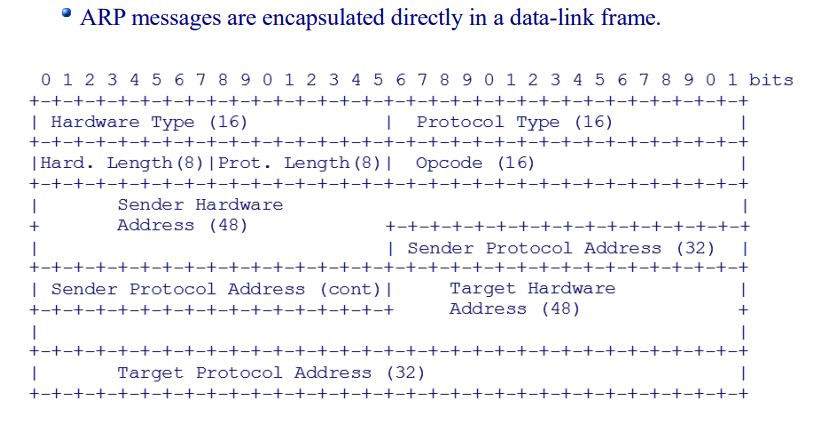
Messages, example:



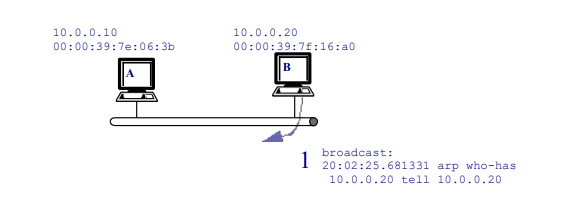
ARP resolution in an ethernet network (broadcast network):

* A broadcast “ARP Request” message is sent indicating the IP address.
* The station having the requested IP address sends a unicast “ARP Reply”, and stores the requesting address in the ARP address.
* Upon receiving the “ARP Reply”, the requesting station return the IP call with it.
* ARP entries have a timeout refreshed each time a match occurs.

Message format (ethernet):



Gratuitous ARP:



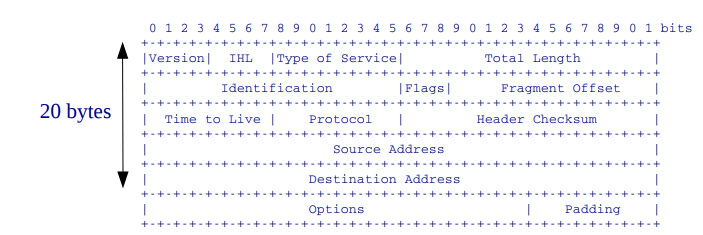
Goals:

* Detect duplicated IP addresses.
* Update MAC addresses in ARP tables after an IP or NIC change.

**IP header (RFC 791)**

Characteristics:

* Version: 4.
* IP Header Length (IHL): Header size in 32-bit words.
* Type of Service (ToS): xxxdtrc0.
* Total Length: Datagram size in bytes.
* Identification / Flags / Fragments Offset: used in fragmentation.
* Time to Live (TTL): if (--TTL == 0) { discard; }.
* Protocol: Encapsulated protocol (/etc/protocols in Unix).
* Header Checksum: Header error detection.
* Source and Destination Addresses: End node addresses.
* Options: Rcord Route, Loose Source Routing, Strict Source Routing.



IP Fragmentation:

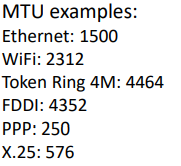
Fragmentation may occur:

* Router: Fragmentation may be needed when two networks with different Maximum Transfer Unit (MTU) are connected.
* Host: Fragmentation may be needed using UDP. TCP segments are <= MTU.

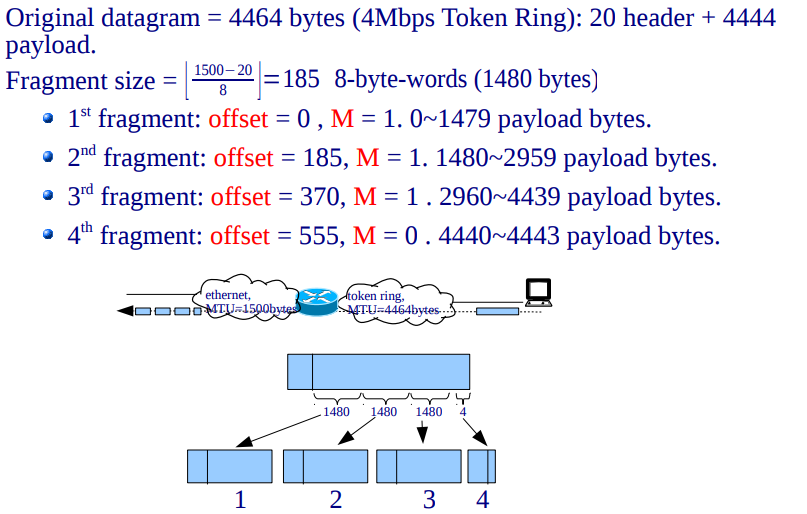
Datagrams are reconstructed at the destination.

Fields:

* Identification (16 bits): identify fragments from the same datagram.
* Flags (3 bits):
  + D, don’t fragment. Used in MTU path discovery.
  + M, more fragments: set to 0 only in the last fragment.
* Offset (13 bits): position of the fragment first byte in the original datagram in 8-byte words (indexed at 0).



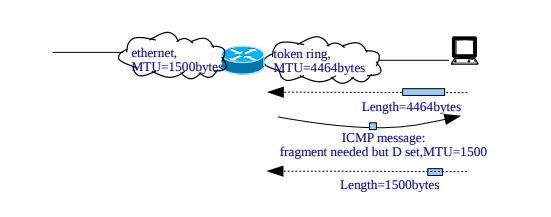
Example:



MTU Path Discovery:

Used in modern TCP implementations. TCP by default chooses the maximum segment size, to avoid headers overhead (segment efficiency = TCP payload / (TCP payload + Σ TCP, IP, Data-link, Physical Headers).

Goal: avoid fragmentation, the DF flag is set to one, segment size is reduced upon receiving ICMP error message “fragmentation needed but DF flag set”.

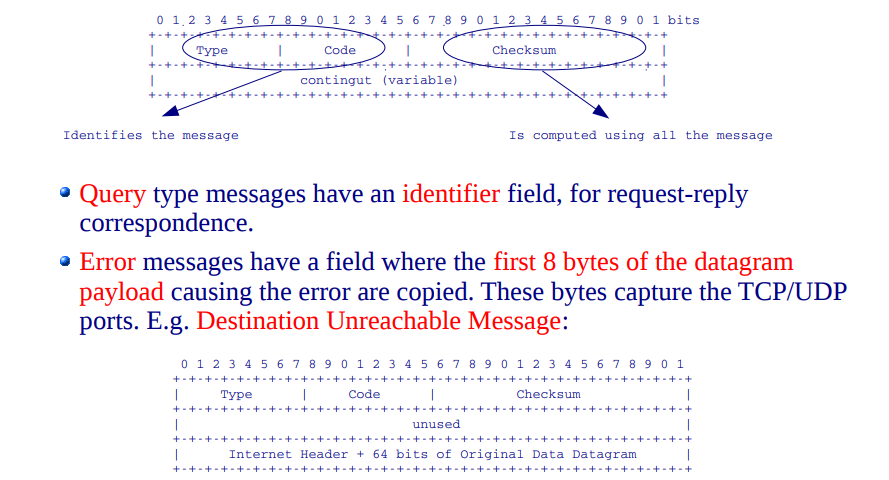


**ICMP protocol**

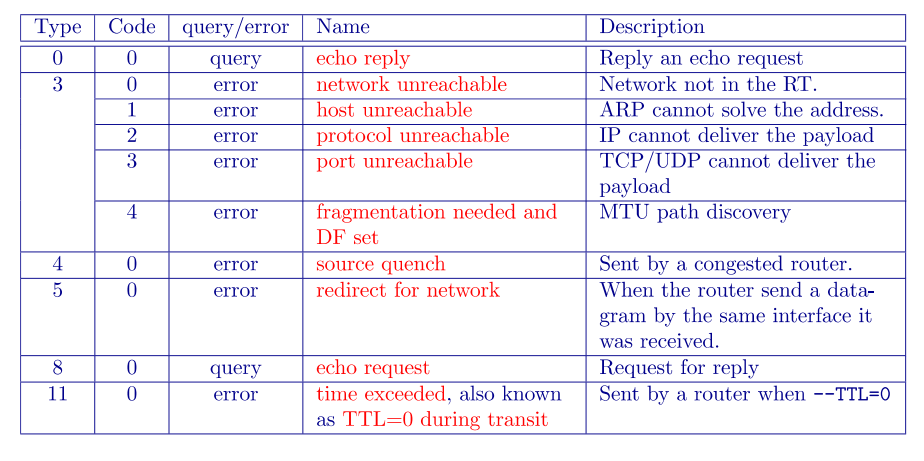
Internet Control Message Protocol, ICMP (RFC 792 Y1981):

* Used for attention and error messages.
* Can be generated by IP, TCP/UDP, and application layers. ICMP.
* Messages are encapsulated into an IP datagram (protocol = 1).
* Messages can be: (i) query, (ii) error.
* An ICMP error message cannot generate another ICMP error message (to avoid loops).

ICMP general format message:



Common ICMP messages:



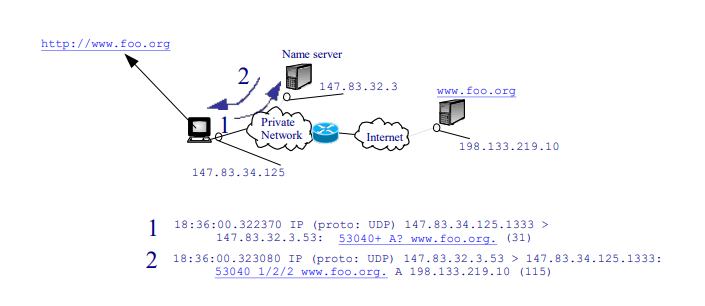
**DNS**

Domain Name System (RFC 1034, 1035 Y1987):

* Allows users to use names instead of IP addresses: e.g., rogent.ac.upc.edu instead of 147.83.31.7, www.upc.edu instead of 147.83.194.21, etc.
* Names consist of a node-name and a domain-name: rogent.ac.upc.edu, www.upc.edu.
* DNS consists of a worldwide distributed data base.
* DNS data base entries are referred to as Resource Records (RR).
* The information associated with a name is composed of 1 or more RRs.
* Names are insensitive (doesn’t matter if it is uppercase or lowercase).

DNS - protocol:

* Client-server paradigm.
* UDP/TCP. For short messages it uses UDP.
* Well-known port: 53.

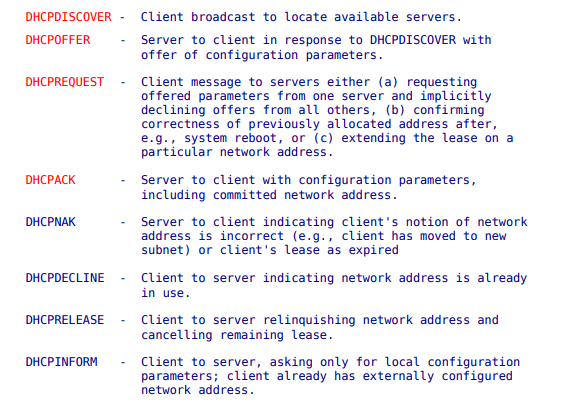


**DHCP protocol**

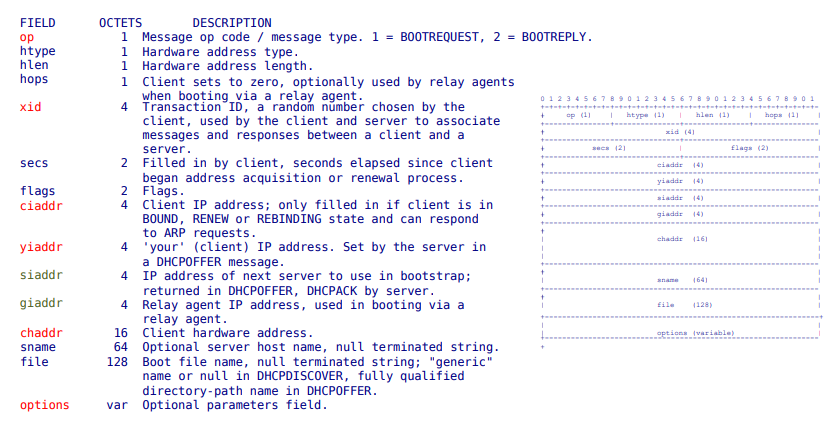
Dynamic Host Configuration Protocol, DHCP (RFC 2131 Y1997):

* Improves and can interoperate with previous BOOTP protocol.
* Used for automatic network configuration:
  + Assign IP address and mask.
  + Default route.
  + Configure DNS servers.
  + Hostname.
  + DNS domain.
* IP addresses configuration can be:
  + Dynamic: during a leasing time.
  + Automatic: unlimited leasing time.
  + Manual: IP addresses are assigned to specific MAC addresses.

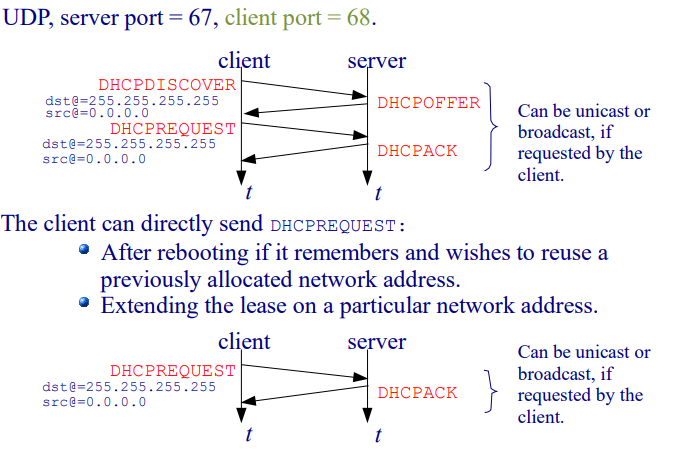
Protocol Messages:

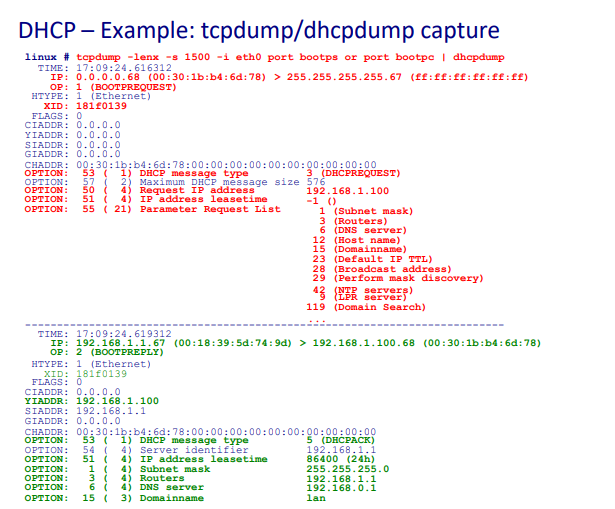


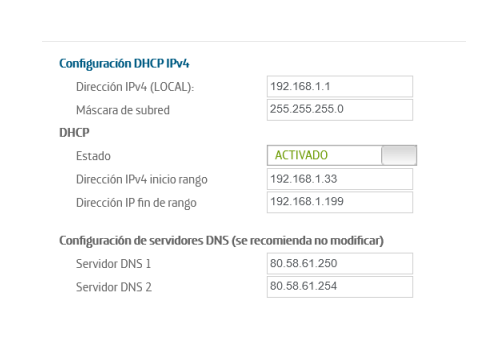
Message Fields:

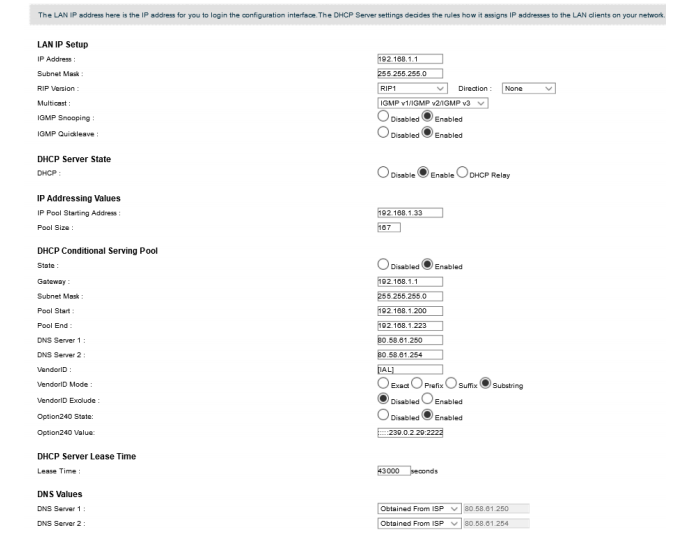


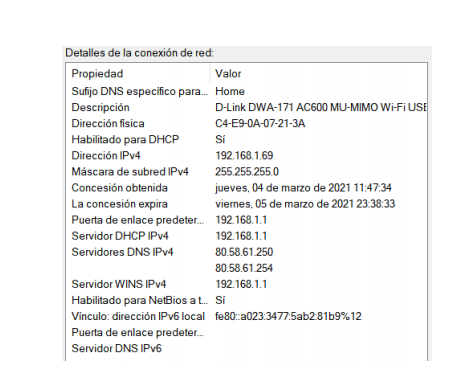
Client-server interaction:







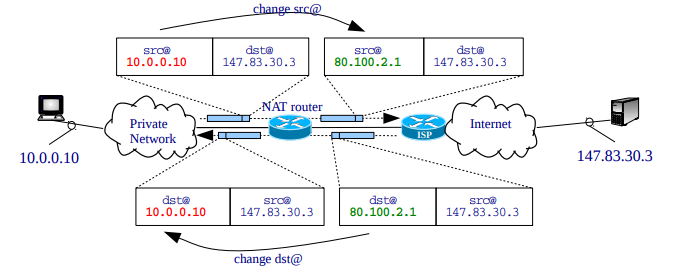




**NAT**

Network Address Translation, NAT (RFCs 1631, 2663, 3022):

* Typical scenario: private addresses (internal addresses) are translated to public addresses (external addresses).
* A NAT table is used for address mapping.
* Advantages:
  + Save public addresses.
  + Security.
  + Administration, e.g., changing ISP does not imply changing private network addressing.

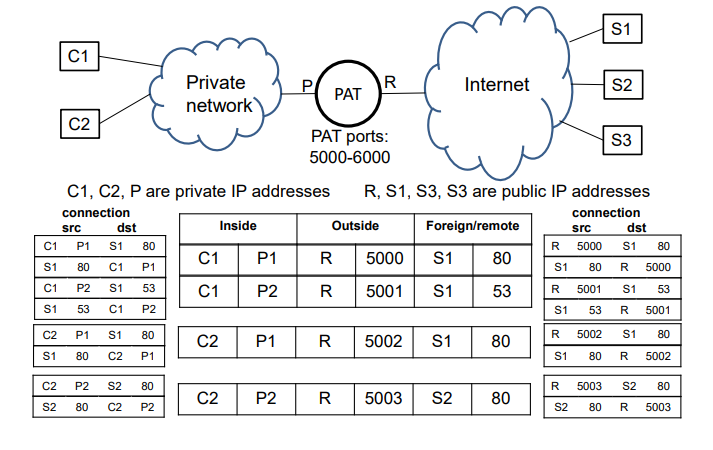


Types of translations:

(\*) NAT is a technique, not a protocol. Implementations and terminology may change from one manufacturer to another.

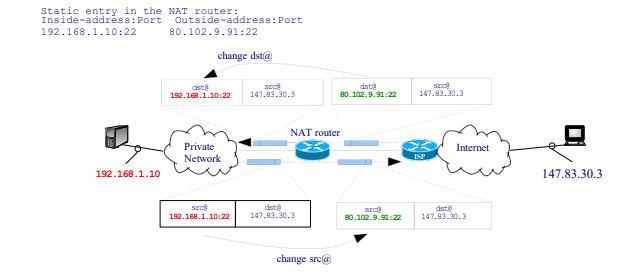
* Basic NAT:
  + A different external address is used for each internal address => a different public IP address is needed for each host accessing Internet.
  + Each NAT table entry has the tuple: (internal address, external address).
  + Each host requires one NAT table entry.
* Port and Address Translation, PAT (PNAT, NAPT):
  + The same external address can be used for each internal address => a unique public IP address can be used for all hosts accessing Internet.
  + Each NAT table entry has the tuple: (int. address/port, ext. address/port).
  + Each connection requires one NAT table entry.
* The NAT table entries can be:
  + Static: Manually added.
  + Dynamic:
    - Entries are automatically added when an internal connection is initiated.
    - External addresses are chosen from a pool.
    - Table entries have a timeout.

Port and Address Translation (PAT):



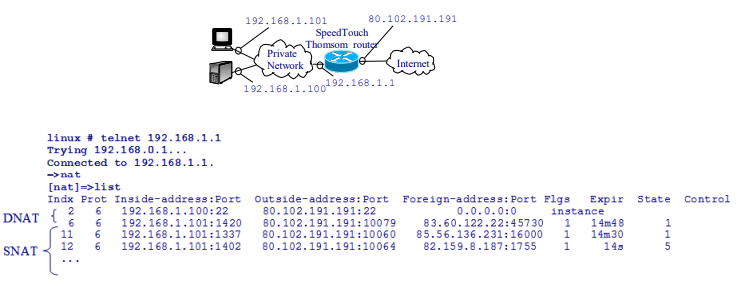
DNAT (Destination NAT):

* What if we want external connections to internal servers? (DNAT in Linux-iptables terminology).
* The address translation is exactly the same as NAT, but the connection is initiated from an external client.
* Typically, some static configuration is needed to configure the server IP/port.



ADSL commercial router example:

* NAT outgoing packets to 80.102.191.191.
* DNAT incoming packets, port 22 (ssh) to 192.168.1.100.



**Routing algorithms**

**Security in IP**